Parameters for Epistemic Gossip Problems

Hans Van Ditmarsch, Davide Grossi, Andreas Herzig, Wiebe van Der Hoek, Louwe B. Kuijer

Parameters for Epistemic Gossip Problems

Hans van Ditmarsch* Davide Grossi† Andreas Herzig‡
Wiebe van der Hoek† Louwe B. Kuijer†

Abstract

We introduce a framework that can model different kinds of epistemic gossip problems. We also formalize some parameters that distinguish the different types of gossip problem. Finally, we present a few results that show the effect of the parameters.

1 Introduction

The so-called *gossip problem* [11, 7, 8] can be described as follows: there are n agents and each of them knows a secret. In the beginning, each agent knows only their own secret. The agents then start making phone calls to each other. Whenever one agent calls another, they tell each other all the secrets they know—including their own secret as well as all secrets that they learned prior to the current call. The goal is to make sure that all agents learn all secrets, and to use as few calls as possible in the process. The gossip problem is of interest as an abstract puzzle, but it also has implications for more practical problems related to information gathering.

Originally, the gossip problem was studied mostly from a combinatorial and graph-theoretical point of view. See [8] for a survey of the combinatorial results about the gossip problem. For our current purpose, the most important combinatorial result is that, if $n \geq 4$, the optimal solutions to the gossip problem require $2n - 4$ calls. More recently, people have started to study the gossip problem from a *knowledge based* or *epistemic* point of view [2, 3, 1].

The difference between the two approaches lies in the perspective they take. In the combinatorial approach, we look at protocols through the eyes of an all-knowing scheduler who tells the agents whom they have to call, and when they should do so. In the epistemic approach, we look at protocols through the eyes of the individual agents, who must make a decision about which call to make based on the information that they have available.

Here we follow the epistemic approach, so the agents base their decision whom to call on what they know. This, of course, makes it important to specify

*Laboratoire lorrain de recherche en informatique et ses applications
†University of Liverpool
‡Institut de Recherche en Informatique de Toulouse

what information is available to the agents and what it means for agents to act on this information (cf. knowledge-based programs [5]). For example, we need to specify whether the system is synchronous, so whether the agents know how many calls have already been made. The usual approach is to choose one answer to this and other questions. In this case, however, we leave these questions open and treat them as parameters. This allows us to study many different gossip problems in one framework.

This brings us to our goals in this paper. First, we develop a framework for gossip problems that is general enough to encompass many of the different parameters. Then, we formally define some of the parameters in this framework. Finally, we prove a few results about the consequences of choosing certain values for the parameters. Since our main goal is to develop a framework for gossip problems, this paper will be rather heavy on definitions. Please bear with us, it can't be helped in this situation.

2 The Formal Framework

The choice of framework requires a balance between generality and usefulness. The more general we make the framework, the more parameters we can model. But this generality comes at the cost of a more unwieldy and complicated system. In order to achieve the desired balance there are a few important things that we will treat as invariant truths, instead of parameters. Firstly, we model the gossip problem on complete graphs, instead of the problem on general graphs or the dynamic gossip problem [4]. This means that we assume that for every two agents a and b, it is always possible for a to call b. So a "knows b's phone number." Secondly, we assume that agents have perfect memory. This does not mean that once an agent knows φ it will always know φ in the future. It does mean that if a used to know φ, then it knows that it used to know φ. Finally, we assume that calls are made sequentially. These assumptions can be weakened by making small changes to the framework, but for reasons of simplicity we will not make these changes in this paper.

In order to reason about knowledge and protocols, we use a variant of Propositional Dynamic Logic [10, 6] that includes a few epistemic operators. Specifically, the language we use is defined as follows.

Definition 1. *Let $\mathcal{A}$ be a finite set of agents and let $\mathcal{P}$ be a countable set of propositional variables such that for every $a, b \in \mathcal{A}$ there is an element $k_{ab} \in \mathcal{P}$. Furthermore, let $\mathcal{C} := \{c_{ab} \mid a, b \in \mathcal{A}, a \neq b\}$. The sets $\mathcal{L}$ of formulas φ and Π of protocols π are given by the following normal forms:*

$$\varphi ::= p \mid \neg\varphi \mid \varphi \vee \varphi \mid K_a\varphi \mid C\varphi \mid [\pi]\varphi$$
$$\pi ::= c_{ab} \mid \varphi? \mid \pi \sqcup \pi \mid \pi;\pi \mid \pi^* \mid \pi^{-1}$$

where $p \in \mathcal{P}$, $a, b \in \mathcal{A}$ and $c_{ab} \in \mathcal{C}$.

We use $\wedge, \rightarrow, \leftrightarrow, \hat{K}_a, \langle\pi\rangle, \bigwedge, \bigvee$ and $\bigsqcup$ in the usual way as abbreviations. Furthermore, we also define $E\varphi := \bigwedge_{a\in\mathcal{A}} K_a\varphi$ and $\mathit{Init} := \bigwedge_{a\neq b} \neg k_{ab} \wedge \bigwedge_a k_{aa}$

as abbreviations. Finally, by convention we let the empty sequence of calls represent $\top?$, so in the degenerate case $n = 0$ the sequence $c_{a_1b_1}; \cdots ; c_{a_nb_n}$ represents $\top?$.

We use c_{ab} to represent the action "agent a calls agent b." We don't allow an agent to call itself, so $\mathcal{C} = \{c_{a,b} \mid a, b \in \mathcal{A} : a \neq b\}$. Furthermore, if $a, b \in \mathcal{A}$ we use $k_{ab} \in \mathcal{P}$ to represent "agent a knows agent b's secret." The reason we use k_{ab} instead of, say, $K_a s_b$ is that $K_a\varphi$ means that a knows that φ is true. Secrets are neither true nor false, so they cannot be known in the K_a sense.

Note that while $\mathcal{P}$ is required to include k_{ab} for every $a, b \in \mathcal{A}$ it also contains other propositional variables. These other variables can be used to encode relevant information. For example, if an agent bases its next call on a coin toss, we can use $p \in \mathcal{P}$ to represent the outcome of the toss.

The agents' goal is to have everyone know all secrets. We define the formula *Goal* as a representation of this goal in the object language, i.e. $\mathit{Goal} := \bigwedge_{a,b} k_{ab}$.

This language can be evaluated on Kripke models with two sets of relations: an indistinguishability relation $R(a)$ for every agent a and an outcome relation $O(c_{ab})$ for every distinct a, b.

Definition 2. *A* model M *is a tuple* $M = (S, R, O, V)$ *where* S *is a set of states,* $R : \mathcal{A} \to 2^{S\times S}$ *assigns to each agent an equivalence relation on* S*,* $O : \mathcal{C} \to 2^{S\times S}$ *assigns to each call a relation on* S *and* $V : \mathcal{P} \to 2^S$ *is a valuation.*

The language $\mathcal{L}$ is evaluated on these models in the usual way.

Definition 3. *Let* $M = (S, R, O, V)$ *be a model and* $s \in S$*. The satisfaction relation* $\models$ *is given inductively by*

$$
\begin{array}{lcl}
M, s \models p & \Leftrightarrow & s \in V(p), \text{ for } p \in \mathcal{P} \\
M, s \models \neg\varphi & \Leftrightarrow & M, s \not\models \varphi \\
M, s \models \varphi_1 \vee \varphi_2 & \Leftrightarrow & M, s \models \varphi_1 \text{ or } M, s \models \varphi_2 \\
M, s \models K_a\varphi & \Leftrightarrow & M, s' \models \varphi \text{ for all } s' \text{ such that } (s, s') \in R(a) \\
M, s \models C\varphi & \Leftrightarrow & M, s' \models \varphi \text{ for all } s' \text{ such that } (s, s') \in \left(\bigcup_{a\in\mathcal{A}} R(a)\right)^*, \\
 & & \text{where } {}^* \text{ indicates the reflexive transitive closure} \\
M, s \models [\pi]\varphi & \Leftrightarrow & M, s' \models \varphi \text{ for all } s' \text{ such that } (s, s') \in O(\pi)
\end{array}
$$

and

$$
\begin{array}{lcl}
(s, s') \in O(\varphi?) & \Leftrightarrow & s = s' \text{ and } M, s \models \varphi \\
(s, s') \in O(\pi_1 \sqcup \pi_2) & \Leftrightarrow & (s, s') \in O(\pi_1) \text{ or } (s, s') \in O(\pi_2) \\
(s, s') \in O(\pi_1; \pi_2) & \Leftrightarrow & \text{there is an } s'' \text{ such that} \\
 & & (s, s'') \in O(\pi_1) \text{ and } (s'', s') \in O(\pi_2) \\
(s, s') \in O(\pi^*) & \Leftrightarrow & (s, s') \in O(\pi)^*, \text{ where } {}^* \text{ indicates the} \\
 & & \text{reflexive transitive closure} \\
(s, s') \in O(\pi^{-1}) & \Leftrightarrow & (s', s) \in O(\pi).
\end{array}
$$

We write $M \models \varphi$ *if* $M, s \models \varphi$ *for all* $s \in S$*.*

We can coherently evaluate $\mathcal{L}$ on all models. In general, models do not accurately represent the gossip problem, however. For example, there is nothing in the definition of a model to guarantee that every agent starts out knowing its own secret, or that agents exchange secrets during calls. We therefore define the subclass of *gossip models*, that model the gossip problem. But before defining gossip models, we first need a few auxiliary definitions.

Definition 4. *Let $M = (S, R, O, V)$ be a model and let $s, s' \in S$. The state s' is a* predecessor *of s if there are $a, b \in \mathcal{A}$ such that $(s, s') \in O(c_{ab})$. The state s is an* initial state *if it has no predecessors.*

Definition 5. *Let $M = (S, R, O, V)$ be a model and let $s \in S$. If there are unique finite sequences $c_{a_1b_1}; \cdots ; c_{a_kb_k}$ of calls and $s_1, \cdots, s_{k+1}$ of states such that*

- *$s_{k+1} = s$ and $(s_m, s_{m+1}) \in O(c_{a_mb_m})$ for all $1 \leq m \leq k$,*
- *s_1 is an initial state*

then $p(s) = c_{a_1b_1}; \cdots ; c_{a_kb_k}$ is the past *of s and $h(s) = s_1, \cdots, s_{k+1}$ is the* history *of s. If there is no such unique sequence, then $p(s)$ and $h(s)$ are undefined.*

If $p(s)$ and $s(s)$ exist, then the state s_1 is the origin state *of s, and k is the* length *of $p(s)$.*

Intuitively, $p(s)$ is the sequence of calls that have already taken place in s, and $h(s)$ is the sequence of states that preceded s. Unfortunately, in general models such sequences are not guaranteed to be well-defined, because the relations $O(c_{ab})$ may be cyclical and because a world may have multiple predecessors. Gossip models are defined in such a way that every state s does have a (by definition unique) past $p(s)$ and history $h(s)$.

Definition 6. *Let $M = (S, R, O, V)$ be a model and let $s \in S$. Suppose $p(s) = c_{a_1b_1}; \cdots ; c_{a_kb_k}$ and let $a \in \mathcal{A}$. Then the* a-reduction *of $p(s)$, denoted $p(s)|_a$, is the subsequence of $p(s)$ that contains exactly those terms $c_{a_mb_m}$ where $a_m = a$ or $b_m = a$.*

So $p(s)|_a$ is the sequence of calls that a participated in.

Definition 7. *Let $M = (S, R, O, V)$ be a model and let $s, t \in S$ and $a \in \mathcal{A}$. Suppose $h(s) = s_1; \cdots ; s_k, s_{k+1}$ and $h(t) = t_1; \cdots ; t_m, t_{m+1}$. Then $h(s)$ and $h(t)$ are* a-similar*, denoted $h(s) \cong_a h(t)$, if there are functions $f_1 : \{1, \cdots, k+1\} \to \{1, \cdots, m+1\}$ and $f_2 : \{1, \cdots, k+1\} \to \{1, \cdots, m+1\}$ such that*

1. *f_1 and f_2 are increasing,*
2. *for every $1 \leq i \leq k+1$, $(s_i, t_{f_1(i)}) \in R(a)$,*
3. *for every $1 \leq i \leq m+1$, $(t_i, s_{f_1(i)}) \in R(a)$,*

If $h(s)$ and $h(t)$ exist, then $h(s) \not\cong_a h(t)$ exactly if a could use its memory to distinguish between s and t. For example, agent a could reason "I used to be in state s_i, or some state indistinguishable from s_i. But there is no state t_j in the history of t that is indistinguishable from s_i. So I cannot be in t."

Definition 8. *A* gossip model M *is a model* $M = (S, R, O, V)$ *such that:*

1. *if* s *is an initial state then for all* $a, b \in \mathcal{A}$ *we have* $s \in V(k_{ab})$ *if and only if* $a = b$*;*
2. *if* $(s, s') \in O(c_{a_1 a_2})$*, then* $s' \in V(k_{a_3 a_4})$ *if and only if* $s \in V(k_{a_3 a_4})$ *or* $(a_3 \in \{a_1, a_2\}$ *and* $(s \in V(k_{a_1 a_4})$ *or* $s \in V(k_{a_2 a_4})))$*;*
3. *if* s *is a non-initial state, then it has exactly one predecessor* s' *and there is exactly one pair* (a, b) *such that* $(s, s') \in O(c_{ab})$*;*
4. *for every* s *and every* $a \neq b$ *there is exactly one* s' *such that* $(s, s') \in O(c_{ab})$*;*
5. *if* $(s, s') \in R(a)$ *then* $s \in V(k_{ab})$ *if and only if* $s' \in V(k_{ab})$*;*
6. *if* $(s, s') \in R(a)$ *then* $p(s)|_a = p(s')|_a$*;*
7. *if* $(s, s') \in R(a)$*, then* $h(s) \cong_a h(s')$*.*

Given a class $\mathfrak{M}$ *of gossip models we write* $\mathfrak{M} \models \varphi$ *if* $M \models \varphi$ *for all* $M \in \mathfrak{M}$*. If* $\mathfrak{M}$ *is the class of all gossip models we write* $\models \varphi$ *for* $\mathfrak{M} \models \varphi$*.*

Definition 8 is quite complicated, so let us explain which aspect of the gossip problem is encoded in which requirement in the definition.

In the gossip problem every agent starts out knowing its own secret, and only its own secret. This is encoded in condition 1. Agents only learn secrets by being involved in calls, and in a call they exchange all secrets they know. This is encoded in condition 2.

We assume that any uncertainty in a gossip setting is epistemic in nature. So in a given state s it is fully determined which calls have taken place in the past, and given an action c_{ab} it is determined what the outcome of performing c_{ab} in s would be.[1] In Definition 8 this is encoded as conditions 3 and 4.

The last three conditions constrain the indistinguishability relations of the agents. Recall that we modeled a knowing b's secret as the propositional variable k_{ab}. This means that introspection about the knowledge of secrets is not guaranteed by the fact that $R(a)$ is an equivalence relation. We use condition 5 to guarantee that agents are introspective about which secrets they know.

Condition 6 demands that the agents remember the calls that they themselves participated in. Finally, condition 7 makes sure that if an agent a is uncertain about whether it is in s_1 or in s_2, then the histories of s_1 and s_2 are consistent, at least as far as a is concerned. So a cannot dispel its uncertainty simply by checking which of the two histories is consistent with its memory.

[1] Note that while the past and the outcomes of actions are fixed, this does not imply that the agents know the past or the outcome of actions.

At this point, we should note that Definition 8 is stronger than what we need for the proofs presented here. We could have omitted conditions 6 and 7 and weakened some of the other conditions without invalidating any of the results that we present. Typically, we would use definitions that are as weak as possible, since this provides the most general results. In this case, however, one of our goals is to design a general framework to reason about different kinds of epistemic gossip problems. As such, we add not just the conditions that we need for our results, but also the other conditions that we believe to be reasonable for a model of the gossip problem.

3 Modeling the Parameters

Our main goal is to provide a framework that can be used to formalize the parameters that distinguish different kinds of epistemic gossip problem. We cannot discuss every parameter here, so we restrict ourselves to two important ones. The first of these parameters is the nature of the protocols that we consider: all protocols, epistemic protocols or symmetric epistemic protocols. The second parameter is the amount of knowledge agents have about calls that they are not directly involved in: is communication asynchronous, synchronous or observable?

Synchrony and observability are not new concepts, of course. But their formalization in the gossip framework is new. Epistemic protocols have already been defined in the context of gossip protocols [3, 2], but the distinction between epistemic and symmetric epistemic protocols is new.

3.1 Constraints on Protocols

In principle, every PDL protocol can be considered on a gossip model. But in general such protocols may require agent a to make a call c_{ab} even if a doesn't know that this call should be made. We are therefore mostly interested in two subclasses of protocols, the *epistemic* and *epistemic symmetric* protocols. We follow [3] and [2] in the definition of epistemic protocols.

Definition 9. *A protocol π is an* epistemic protocol *if it is of the form*

$$\pi = \left(\bigsqcup_{a \neq b \in \mathcal{A}} K_a \psi_{ab} ?; c_{ab} \right)^* ; \left(\neg \bigvee_{a \neq b \in \mathcal{A}} K_a \psi_{ab} \right) ?$$

where ψ_{ab} is a formula for every $a \neq b \in \mathcal{A}$. The formula $K_a \psi_{ab}$ is called the call condition *for c_{ab} in π.*

Many constructions in PDL have abbreviations that resemble the way one would phrase a protocol in a programming language. Using such an abbreviation, an epistemic protocol π can be written as

$$\text{while } \bigvee_{a \neq b} K_a \psi_{ab} \text{ do } \bigsqcup_{a \neq b} (\text{if } K_a \psi_{ab} \text{ then make call } c_{ab}).$$

So as long as at least one of the call conditions is true, the protocol nondeterministically executes one of the actions c_{ab} for which the call condition holds. We allow only one call condition per pair a, b, but we can always combine multiple call conditions into one. If, for example, we want to allow a call c_{ab} if either $K_a\varphi_{ab}$ or $K_a\chi_{ab}$, this is equivalent to allowing the call c_{ab} if $K_a(K_a\varphi_{ab} \vee K_a\chi_{ab})$.

In some cases we want to restrict the available protocols even further, to the symmetric epistemic protocols. A symmetric protocol is one where all agents are treated in the same way. In other words, a protocol is symmetric if all agents "are running the same software." So, for example, a protocol is not symmetric if it assigns one pre-determined agent a the role of leader, and requires this a to call every other agent.

Definition 10. *A* permutation *on $\mathcal{A}$ is a bijection $\sigma : \mathcal{A} \to \mathcal{A}$. Given a permutation σ and a formula φ, the formula $\sigma(\varphi)$ is obtained by simultaneous replacement of all occurrences in of a in φ by $\sigma(a)$ for all $a \in \mathcal{A}$.*

Definition 11. *Let π be an epistemic protocol, and for every $a \neq b \in \mathcal{A}$ let $K_a\psi_{ab}$ be the call condition for c_{ab}. The protocol π is* symmetric *if for every permutation σ on $\mathcal{A}$ and every $a \neq b \in \mathcal{A}$, we have $\sigma(\psi_{ab}) = \psi_{\sigma(a)\sigma(b)}$.*

3.2 Knowledge of Other Agents' Calls

In our gossip models we assume that agents are aware of all calls that they themselves participate in, so if a cannot distinguish between state s and s' then $p(s)|_a = p(s')|_a$. In some cases we may also want to make assumptions about agents' knowledge of calls they do not participate in. We distinguish three levels of such knowledge.

Firstly, it is possible that the agents are simply unaware of calls that they do not participate in. In this case, the setting is *asynchronous*. Secondly, it is possible that the agents know how many calls take place, but don't know who is participating in the calls (unless they are one of the participating agents). In this case, the setting is *synchronous*. Finally, it is possible that the agents know exactly which calls take place, although they cannot observe the content of the call (so an agent that is not involved in the call will not learn the values of the secrets that are exchanged). In this case, the setting is *observable*.

The three different levels of knowledge can be represented by constraints on the knowledge relation in gossip models. For the asynchronous case we need not place any restrictions, we can consider the class of asynchronous models to be equal to the class of all gossip models. The classes of synchronous and observable models do require restrictions. The observable models can easily be defined as follows.

Definition 12. *A gossip model $M = (S, R, O, V)$ is* observable *if for all $s, s' \in S$ and all $a \in \mathcal{A}$ such that if $(s, s') \in R(a)$, we have $p(s) = p(s')$. Let $\mathfrak{M}_{Obs}$ be the class of observable models.*

Defining the synchronous models is slightly harder, and requires an auxiliary definition.

Definition 13. *Let $M = (S, R, O, V)$ be a gossip model, $s \in S$ and $a \in \mathcal{A}$. Suppose $p(s) = c_{a_1 b_1}; \cdots ; c_{a_k b_k}$. The* synchronous a-reduction *of $p(s)$, noted $p(s)|_a^{synch}$, is the sequence $x_1; \cdots ; x_k$ where $x_m = c_{a_m b_m}$ if $a_m = a$ or $b_m = a$ and $x_m = \epsilon$ otherwise.*

The symbol ϵ is used as a placeholder for "a call took place, but I don't know which one".

Definition 14. *A gossip model $M = (S, R, O, V)$ is* synchronous *if, for every $s, s' \in S$ and all $a \in \mathcal{A}$ such that $(s, s') \in R(a)$, we have $p(s)|_a^{synch} = p(s')|_a^{synch}$. Let $\mathfrak{M}_{Synch}$ be the class of synchronous models.*

4 Optimal Solution Length

Now that we have defined some of the parameters for gossip protocols, we can determine how the parameters affect the gossip problem. There are several aspects of the gossip problem that can change based on the parameters. For example, in some cases we can get common knowledge of *Goal* while in other cases this is impossible. The aspect of the gossip problem that we study here is closer to the traditional questions about gossip: we want to determine how many calls are required to achieve *Goal* in the different scenarios.

Definition 15. *Let $\mathfrak{M}$ be a class of models. A protocol π is* executable *on $\mathfrak{M}$ if for all $M \in \mathfrak{M}$ and every initial state s of M we have $M, s \models \langle \pi \rangle \top$. A protocol π is* effective *on $\mathfrak{M}$ if for all $M \in \mathfrak{M}$ and every initial state s of M, we have $M, s \models [\pi] Goal$. Given $m \in \mathbb{N}$, a protocol π is m-*effective *on $\mathfrak{M}$ if it is effective on $\mathfrak{M}$ and furthermore, for every $M \in \mathfrak{M}$ and every initial state s of M, all traces of π in M, s are of length at most m.*

Definition 16. *Let $\mathfrak{M}$ be a class of models and Γ a set of protocols. The* optimal solution length *for Γ on $\mathfrak{M}$ is the lowest $m \in \mathbb{N}$ such that there is a protocol $\pi \in \Gamma$ that is executable and m-effective on $\mathfrak{M}$.*

We have not yet fully determined the optimal solution length for all combinations of models and protocols discussed above. The ones that we do know, as well as bounds for the ones that we have not yet fully determined, are as follows (assuming $n \geq 4$, and in the asynchronous/symmetric epistemic case that $n \geq 8$ and n is even).

	Observable	Synchronous	Asynchronous
All protocols	$2n-4$	$2n-4$	$2n-4$
Epistemic	$2n-4$	$2n-4$	$\leq 2n-3$
Symmetric Epistemic	$2n-4$	$2n-3$	$\geq 2n$

We do not yet have a proof for the exact optimal solution lengths on asynchronous models, but we conjecture that the optimal solution length on asynchronous models is $2n-3$ for epistemic protocols and in $O(n^2)$ for symmetric epistemic protocols. We do not provide full proofs for these optimal solution

lengths here, but we do give proof sketches for all but one of the non-trivial cases. In the proof sketches we informally describe a number of protocols. These protocols can be formalized in a straightforward way, but such formalizations are harder to read than the informal description. We therefore omit the formalizations.

Observable models. In observable models all agents know the call history. Consider the protocol where a call c_{ab} is allowed if and only if a knows that the current history followed by c_{ab} is an initial segment of some effective sequence of calls with length $2n-4$. This is an epistemic symmetric protocol, and it is executable and $(2n-4)$-effective on the class of observable models.

Synchronous models. In synchronous models all agents know how many calls have taken place. Fix any call sequence that causes all agents to know all secrets after $2n-4$ calls. We can then require all agents to take turns creating this call sequence: if the k-th call in the sequence is c_{ab}, then agent a should make that call after $k-1$ other calls have happened. This is an epistemic protocol, but not a symmetric one.

For the lower bound of the synchronous/symmetric case we need a combinatorial property that holds for any call sequence of $2n-4$ calls that is effective: there are at least two calls c_{ab} and $c_{a'b'}$ between "fresh" agents that had not participated in any calls before. After all, suppose towards a contradiction that there is only one call between fresh agents. Then in the first call the number of fresh agents decreases by 2, and every call afterwards it decreases by at most 1. So after $n-2$ calls there is at least one fresh agent left. Consider the number of agents that know this last fresh agent's secret. After $n-2$ calls, only the agent itself knows it. Every call afterwards can teach it to at most one more agent, so it takes at least $n-1$ more calls for every agent to know this secret. In total, the call sequence therefore has to contain at least $2n-3$ calls.

This property of having at least two calls between fresh agents cannot be guaranteed by an epistemic symmetric protocol in a synchronous model; agents know how many calls have taken place but not who was involved in the calls so they cannot decide to call an agent that has not been called yet. As such, the optimal solution length for symmetric epistemic protocols on synchronous models is at least $2n-3$.

To see that $2n-3$ is also an upper bound, consider the following protocol: every agent starts out by trying to call every other agent. One of the agents, say agent a_1, will be the first one to successfully place a call. On synchronous models, all agents know that the first call has taken place. While most agents do not know who placed the first call, they do know that it wasn't them. So after the first call, continue by having everyone but a_1 remain passive while a_1 calls all other agents. Then, after a_1 has called all agents and therefore knows all secrets, a_1 calls every agent again, except for the agent that was called last in the first round (because that agent already knows all secrets). This protocol is symmetric epistemic, and it is $(2n-3)$-effective on synchronous models.

Asynchronous models. We are not yet certain whether an epistemic $(2n-4)$-effective protocol exists on asynchronous models, although we suspect it does not. We do know that there is an epistemic $(2n-3)$-effective protocol:

fix one agent a. Like in the symmetric epistemic/synchronous case, have a call all other agents and then all but one of the other agents again. We fixed one particular agent a, so this protocol is not symmetric. It is epistemic, however.

This leaves only the asynchronous/epistemic symmetric case. The proof for the $2n$ lower bound in this case is rather long and not very insightful. As such, we omit it here.

5 Conclusion

We introduced a framework that is capable of modeling many different kinds of epistemic gossip problem. We also formally defined some of the parameters for epistemic gossip protocols, and showed the effect of choosing different parameters by considering how they affect the optimal solution length.

One target for future research is the exact optimal solution length on asynchronous models. The case of symmetric epistemic protocols is especially interesting, since there is a relatively large difference between the bound of $2n$ that we can prove and the $O(n^2)$ bound that we believe to hold.

Furthermore, although we have formalized some parameters, there are many more that we have not yet formalized. As such, another interesting target for future research is to treat more of these parameters. One interesting variant is to consider higher-order knowledge: in the variants of the gossip problem that we discussed here, the agents try to make sure that every agent knows all secrets. But there are other variants, where the agents want to make sure that every agent knows that every agent knows every secret, and so on. For the set of all protocols, it was proved in [9] that shared knowledge of depth d, for $d \geq 1$ can be achieved in at most $(d+1) \times (n-2)$ calls. For the smaller sets of protocols, it is not yet known how hard it is to achieve higher order knowledge of secrets.

Other interesting parameters include how much agents tell each other in their conversations (do they only tell each other their secrets, or also when and how they learned the secrets?) and how much the agents know about the protocol that they are collectively following (is the protocol common knowledge?).

Acknowledgments

Some of the parameters discussed here were inspired by discussion at the 2015 Lorentz Center workshop "To Be Announced!" We would like to thank the participants of the workshop for their input in the discussion. We would also like to thank the LOFT reviewers for their remarks and suggestions. Hans van Ditmarsch acknowledges support from ERC project EPS 313360. He is also affiliated to IMSc, Chennai, India, and Zhejiang University, China.

References

[1] Krzyzstof R. Apt, Davide Grossi, and Wiebe van der Hoek. Epistemic protocols for distributed gossiping. In R. Ramanujam, editor, *Proceedings*

of the 15th Conference on Theoretical Aspects of Rationality and Knowledge (TARK 2015), pages 47–55, 2015.

[2] Maduka Attamah, Hans van Ditmarsch, Davide Grossi, and Wiebe van der Hoek. Knowledge and gossip. In T. Schaub, G. Friedrich, and B. O'Sullivan, editors, *Proceedings of the 21st European Conference on Artificial Intelligence (ECAI 2014)*, pages 21–26, 2014.

[3] Maduka Attamah, Hans van Ditmarsch, Davide Grossi, and Wiebe van der Hoek. The pleasure of gossip. In C. Başkent, L. Moss, and R. Ramanujam, editors, *Rohit Parikh on Logic, Language and Society*, To Appear.

[4] Hans van Ditmarsch, Jan van Eijck, Pere Pardo, Rahim Ramezanian, and François Schwarzentruber. Dynamic gossip, 2015. Manuscript. Available at http://arxiv.org/pdf/1511.00867.pdf.

[5] Ronald Fagin, Joseph Y. Halpern, Yoram Moses, and Moshe Vardi. *Reasoning About Knowledge*. MIT Press, Cambridge, MA, USA, 1995.

[6] Michael J. Fischer and Richard E. Ladner. Propositional dynamic logic of regular programs. *Journal of Computer and System Sciences*, 18(2):194–211, 1979.

[7] A. Hajnal, E. C. Milner, and E. Szemerdi. A cure for the telephone disease. *Canadian Mathematical Bulletin*, 15:447–450, 1972.

[8] Sandra M. Hedetniemi, Stephen T. Hedetniemi, and Arthur L. Liestman. A survey of gossiping and broadcasting in communication networks. *Networks*, 18(4):319–349, 1988.

[9] Andreas Herzig and Faustine Maffre. How to share knowledge by gossiping. In *Proceedings of the 3rd International Conference on Agreement Technologies (AT 2015)*, 2016.

[10] Vaughan R. Pratt. Application of modal logic to programming. *Studia Logica*, 39(2):257–274, 1980.

[11] Robert Tijdeman. On a telephone problem. *Nieuw Archief voor de Wiskunde*, 19:188–192, 1971.